REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1, 1997

3. REPORT TYPE AND DATES COVERED
Final Progress Report

4. TITLE AND SUBTITLE
Real-Time Monitoring & Control Methodologies to Aid in the Study of Protein Interactions for Biosensor Applications

5. FUNDING NUMBERS
DAAH04-94-G-0095

6. AUTHOR(S)
Azer Bestavros

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Trustees of Boston University
Cummington Street
Boston, MA 02215

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 32832.4-LS

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

There is an emergent need for portable, inexpensive, reusable, and reliable sensors of environmental contaminants for both military and civilian uses. The development and study of such sensors is crucial for many applications. Among others, we mention measuring groundwater and waste water contamination, monitoring processes for agricultural and food products, and detecting soil contamination. In addition, such a technology is key to many medical procedures. The use of these sensors may be required on a continual and/or on a regular basis. In that respect, optically based sensors of protein interactions—such as those utilizing the optical phenomenon of Surface Plasmon Resonance (SPR)—show promise in producing sensory systems that are reusable, precise, and portable for a range of applications. This proposal is to develop a computing platform to be used in the real-time monitoring, studying, and control of protein interactions. In particular, this platform allows the adaptive control of biosensor devices by integrating various data analysis and decision support modules, which could be used in a real-time fashion to control on-going experiments, or synthesize and schedule new experiments in a distributed environment.

14. SUBJECT TERMS
Real-Time Databases; Biosensor Databases; Scientific Databases; Data Mining; Protein Interactions

15. NUMBER OF PAGES
11 pages

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as; prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NORFORN, REL, ITAR).

DOD - See DoDD 4230.25, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Block 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Real-time Computer Monitoring & Control Methodologies *to Aid in the Study of* Protein Interactions for Biosensor Applications.

FINAL PROGRESS REPORT

AZER BESTAVROS

MARCH 1, 1997

U.S. ARMY RESEARCH OFFICE

DAAH04-94-G-0095

TRUSTEES OF BOSTON UNIVERSITY

OUTLINE OF RESEARCH FINDINGS

1 Introduction

This project aimed at developing a platform that allows the adaptive control of biosensor devices by integrating various data analysis and decision support modules, which could be used in a real-time fashion to control on-going experiments, or synthesize and schedule new experiments in a distributed environment. Due to the reduced level of funding for Year 2, we have decided to concentrate our work on the core component of our platform, namely the biosensor database, while leaving the design and implementation of the data analysis and decision support modules to future years. Our results so far include a detailed architecture of the biosensor database. Amongst the salient features of our architecture is the full support of real-time transactions (with both soft and firm deadlines) and temporal data maintenance. These capabilities are likely to be necessary features of any application using the biosensor database in process control experiments. In the remainder of this report, we discuss these functionalities.

Problem Statement and Motivation Biosensor experiments, like many other scientific experiments, generate a steady stream of data that has to be stored and organized into databases. These databases become powerful tools for designing and scheduling new experiments and for virtual instrumentation. Our investigation of manufacturer-provided biosensor database support for a number of state-of-the-art biosensor equipment (e.g. BIAcore by Pharmacia Biosensor and INTEGRAL by PerSeptive Biosystems) has revealed that these so-called databases are nothing more than unstructured flat files with no support for any database functionalities (e.g. simple queries). These databases are built for specific instruments and for limited purposes; they are generally heterogeneous in terms of semantics, data model, database management system, underlying operating systems, and hardware. As a result, most of them can only be accessed in a "standalone" mode, usually with an ad hoc set of utilities. Moreover, they do not adequately support temporal, spatial, image, sequence, graph, and other structured data. Therefore, we have concluded that there is an urgent need for the development of a more rubust information infrastructure, before developing the rule-based decision subsystems. This need is felt by many scientists in many related disciplines. A report on the importance of Scientific Database Management (prepared for the National Science Foundation) starts with a quote that reads: "[Science initiatives] will founder on the rocks of indifference to data access and information management unless an aggressive and supportive new approach is taken — beginning now."

2 Database Support for Biosensor Applications

Manufacturer-provided biosensor databases have proliferated in such a way that it is not feasible to combine them all into a single global framework because of, among other things, the difficulty in working out a resolution to the heterogeneity and autonomy problems. A practical solution to this problem is offered by the Federated DataBase Systems (FDBS), whereby a common data model is used to integrate multiple schemas of independent databases. FDBSs can be categorized into two types: tighly coupled and loosely coupled. In a tightly coupled FDBS, an intemediate schema is adopted to act as the interface between the local schemas. Tools to transform data from the local schemas to the intermediate schema and to combine these into a federated schema are invoked by application programs. In a loosely copuled FDBS, there is no global schema; the federated system relies on a global language to facilitate cooperation among the member databases. The lack of a central control makes it hard to ensure a global integrity of the FDBS, which results in increased difficulty in meeting the classical requirements for consistency, concurrency, query optimization, and transaction management. Thus, loosely coupled FDBSs are more suitable for systems where updates are infrequent and/or where inconsistencies could be tolerated. Examples of loosely-coupled FDBS include the XBio system of, the Litwin et al system, and the FunBase system.

The database design we have chosen is generic enough to allow easy interfacing to a variety of biosensor equipment and experiments. In that respect, we adopted an object-oriented approach to facilitate data encapsulation and prototyping. An Object Oriented DataBase System (OODBS) promises to accommodate effectively the large and complex data from different instruments, tests, and supporting information. This information may be simple metadata (such as text from a technician, date and time a sample was taken or a test was completed, ... etc.) or complex multimedia data (such as response curves from an instrument, images of the sample ... etc.) An OODBS can more effectively handle the complex interrelations connecting various results and supporting information than traditional DBs. Also, hierarchical relations classifying the objects should lead to higher performance and provide a foundation for a knowledge base to build more "intelligent" instrument control and monitoring systems. Given the unavailability of commercial OODBS or standard that can satisfy our needs, we have started on a collaborative effort with (European Molecular Biology Laboratory) EMBL, whereby our research on issues of data representation and persistance will be incorporated into the SciTools project.

2.1 Support for Real-Time Operation

Our work on adapting current database infrastructures to the real-time constraints that may be imposed on the operation of a biosensor database has focussed on two main problems: Concurrency control and admission control. We discuss our results in both of these below.

Speculative Concurrency Control Various concurrency control algorithms differ in the time when conflicts are detected, and in the way they are resolved. Pessimistic Concurrency Control (PCC) protocols [5, 7] detect conflicts as soon as they occur and resolve them using *blocking.* Optimistic Concurrency Control (OCC) protocols [3, 14] detect conflicts at transaction commit time and resolve them using *rollbacks.*

For a conventional DataBase Management System (DBMS) with limited resources, performance studies of concurrency control methods [2] have concluded that PCC locking protocols perform better than OCC techniques. The main reason for this good performance is that PCC's blocking-based conflict resolution policies result in resource conservation, whereas OCC's restart-based conflict resolution policies waste more resources. While abundant resources are usually not to be expected in conventional database systems, they are more common in real-time environments [6], which are engineered to cope with rare high-load conditions, rather than normal average-load conditions. For example, Real-Time DataBase Systems (RTDBS) are engineered not to guarantee a particular throughput, but to ensure that in the rare event of a highly-loaded system, transactions (critical ones in particular) complete before their set deadlines [4]. This often leads to a computing environment with far more resources than what would be necessary to sustain average loads. In such environments, the advantage that PCC blocking-based algorithms have over OCC restart-based algorithms vanishes. In particular, under such conditions, OCC algorithms become attractive since computing resources wasted due to restarts do not adversely affect performance. Haritsa *et al* [9, 8] investigated the behavior of both PCC and OCC schemes in a real-time environment. The study showed that for a RTDBS with firm deadlines (where late transactions are immediately discarded) OCC outperforms PCC, especially when resource contention is low. The key result of this study is that, if low resource utilization is acceptable (*i.e.* a large amount of wasted resources can be tolerated) then a restart-oriented algorithm that allows a higher degree of concurrent execution becomes a better choice.

Real-time concurrency control schemes considered in the literature could be viewed as extensions of either PCC-based or OCC-based protocols. In particular, transactions are assigned priorities that reflect the urgency of their timing constraints. These priorities are used in conjunction with PCC-based techniques [1, 2, 20, 10, 18, 16, 17] to make it possible for more urgent transactions to abort conflicting, less urgent ones (thus avoiding the hazards of blockages); and are used in conjunction with OCC-based techniques [13, 9, 8, 11, 12, 15, 19] to favor more urgent transactions when conflicting, less urgent ones attempt to validate and commit (thus avoiding the hazards of restarts).

In this project we proposed a categorically different approach to concurrency control that combines the advantages of both OCC and PCC protocols while avoiding their disadvantages. Our approach relies on the use of *redundant* computations to start on alternative schedules, as soon as conflicts that threaten the consistency of the database are detected. These alternative schedules are adopted *only if* the suspected inconsistencies materialize; otherwise, they are abandoned. Due to its nature, this approach has been termed *Speculative Concurrency Control* (SCC). SCC protocols are particularly suitable for RTDBS because they reduce the negative impact of blockages and rollbacks, which are characteristics of PCC and OCC techniques. Our

studies so far have confirmed that SCC protocols provide for a very natural (and elegant) way of incorporating transaction deadline and criticalness information into concurrency control for RTDBS. In particular, SCC protocols introduce a new dimension (namely redundancy) that can be used for that purpose: By allowing a transaction to use more resources, it can achieve better *speculation* and hence improve its chances for a timely commitment. Thus, the problem of incorporating transaction deadline and criticalness information into concurrency control is reduced to the problem of rationing system resources amongst competing transactions, each with a different payoff to the overall system.

Admission Control and Scheduling The proliferation of Real-Time Database (RTDB) Systems as repositories of information used by "time-critical" applications has been tremendous during the last decade. Many such systems continue to admit transactions to the system to the point of overload which results in degraded performance. By the appropriate use of admission control and overload management techniques, the performance of such systems may be enhanced. Moreover, for some safety-critical applications (such as command and control systems), safety constraints require the early notification of transaction failure. Late failure notification is not desirable given that precious system resources, which could have been used by other admitted transactions, are wasted on transactions which end up not completing on-time.

In this project, we designed ACCORD, an Admission Control and Capacity Overload management Real-time Database system, a new framework—a system architecture and a transaction model—for hard deadline RTDB Systems. The system architecture provides for early notification of transaction failure through the use of an admission control mechanism that "prohibits" transactions which are deemed not valuable or incapable of completing on-time from entering the system. The transaction model consists of two components: a primary task and a compensating task. Transactions which are admitted to the system are guaranteed, by the deadline of the transaction, one of two outcomes: either the primary task will successfully commit or the compensating task will safely terminate. Our admission control mechanism permits transactions to fail at the earliest possible point in time (i.e. at submission time) rather than at a later time. Also as a system becomes overloaded, our admission control techniques allow for the utilization of system resources in the most "profitable" way.

The contributions of our work in this particular aspect of our project are: (1) a novel ACCORD framework for RTDB Systems including a system architecture and a transaction model, (2) a value-cognizant admission control mechanisms based upon workload, (3) a value-cognizant admission control mechanisms based upon the level of concurrency conflicts, (4) new scheduling algorithms suitable for ACCORD, and (5) a generalization of ACCORD to broaden its scope to handle soft deadline RTDB Systems. These contributions are validated by an extensive analysis of ACCORD which demonstrates the performance benefits of admission control, overload management, and early failure notification.

2.2 Large-Scale Scientific Databases

The wealth of data available on the internet and the collaborative nature of science today has lead us in the direction of looking at the scientific database issue as distributed and heterogeneous data stores connected via the internet . Even with its current limitations, the internet and the web give a decent foundation. Some items of importance to scientific experiments include: access to remote data, one-to-one and one-to-many communication facilities, web documents can be viewed on all popular computing platforms implementation of a secure, platform-independent programming language (JAVA).

Current internet and web standards are not yet adequate to fully support scientific research on biosensors. Some of the items that are lacking include: access and data security, real-time support for storage and dispersal of data, versioning support for data, functions and experiment models, and context mediation for automatic data conversions. Our current efforts toward enabling scientists to query, analyze, and construct experiments utilizing local and remote data sources are focused on these four issues.

Our approach is to build upon the existing internet standards and to push the limits of the emerging standards towards the needs of scientists. As a base, it is our intention to utilize the Hyper-G server (http://www.tu-graz.ac.at/) developed at Graz University in Austria. This server and related browser enables "UNIX like" security measures for users and groups. Unlike most web servers, the Hyper-G server identitfies the user who is browsing the information. This permits the server to allow access to documents only by those who are authorized. Another benifit of the Hyper-G server is that the links and documents are separate entities, which allows one to personally (or publicly) annotate documents that are read-only. There are a number of other benefits to utilizing a Hyper-G server and browser including, 3D visualization of documents and links, ability to group documents in "clusters" and to have documents be members of more than one cluster, and server level caching scheme for remote documents.

Real-time support for information storage and dispersal will likely be along the line of our earlier work on Adaptive Information Dispersal Algorithm (AIDA). The AIDA approach would add a high degree of security for information dispersal and at the same time improve the expected transmission time of the data.

We believe that versioning support is crucial to a scientific database. The nature of scientific experiments and data collection necessitates collecting and archiving data without knowing what subset of the archived data may be important later. Also, any data analysis that is conducted needs to properly reference the correct data set that was analyzed. Versioning support and a query language (eg. TSQL) for databases which contains versioned data is needed. One approach would be to use the "cluster" facility in Hyper-G to create clusters of versions of the same object. These version clusters would contain some pointers to the "most recent version", "first version", etc. And each document by a single author would contain two pointers, one to the previous version and one to the next version. This approach would not be space optimal and needs to be investigated further.

Maybe the most difficult issue is context mediation or using the more common terms of combining heterogeneous data sets. There are a number of approaches including: having everyone use the same database scheme and format, creating federated databases that all understand the same common (and least rich) format and then a common interface is presented to all who are requesting information, or to teach a database how to communicate with x other databases and interpret data from them. But these approaches may not be adequate. The first approach is not adequate because different applications require different database technologies. The federated approach is time consuming and requires one to relinquish some control of ones database to the federation. And the third approach is extremely time consuming since each database has to know how to communicate with every other database of interest. Another approach would be to use a metadata language to describe the assumptions about the representation and interpretation of the data. This approach is taken by The Context Interchange Project (http://rombutan.mit.edu/context.html) at MIT. That project is a part of ARPA's I3 (Intelligent Integration of Information) Initiative (http://dc.isx.com/I3/).

2.3 Data Mining

There is a wealth of information available through the Internet that may be quite valuable to laboratory scientists. In essence, the Internet complements the local database infrastructure. Therefore, adequate Internet navigation and searching tools must be integrated in any computing platform for laboratory experiments, such as the one we are designing for biosensor applications. To be easily adaptable to using Internet resources (such as the WWW), we intend to use the HTTP protocol and the HTML language as the global communication/language that will glue together the loosely-coupled federated autonomous biosensor databases. Our choice is motivated by the growing popularity of the HTTP/HTML infrastructure, its capability for supporting a wide variety of formats (e.g. pictures, links to metadata, publications, etc), and its ability to provide a homogeneous access to data and programs (e.g. cgi-bin tools, forms, and queries). We believe that a loosely-coupled FDBS is appropriate for our purposes since issues like concurrency control and query optimization are not central (or even applicable) to our system. For example, most data for a biosensor experiment is written once, thereafter it is read-only, thus alleviating the need for concurrency control in such distributed autonomous loosely-coupled systems.

Two completed projects at Boston University dealt with the data distribution issues the biosensor database will have to address. The first focuses on the development and implementation of efficient demand-based data replication protocols in distributed information systems (such as the World Wide Web). The distributed nature and massive volume of the databases to be managed in the proposed CBMCE makes for a perfect testbed for these protocols. The second studies issues related to the design and implementation of distributed scientific and real-time databases, which we describe nect.

LIST OF ALL PUBLICATIONS AND TECHNICAL REPORTS

A. Bestavros and P. Dell. Object-oriented laboratory instrument scheduling and monitoring system: Database issues. In *OOPSLA '94 Workshop on Object-Oriented Technology for Health Care and Medical Information Systems*, October 1994.

A. Bestavros. Speculative algorithms for concurrency control in responsive databases. In Miroslaw Malek, editor, *Responsive Computer Systems.* Kluwer Academic Publishers, 1994.

A. Bestavros. Speculative concurrency control for multidabases with real-time interoperable subsystems. *Journal of Integrated Computer-Aided Engineering*, John Wiley.

Azer Bestavros and Sue Nagy. An Admission Control Paradigm for Real-Time Databases. Technical Report BUCS-TR-96-002, Boston University, Computer Science Department, January 1996.

Azer Bestavros. Timeliness via speculation for real-time databases. Technical Report BUCS-TR-94-008, Computer Science Department, Boston University, Boston, MA, May 1994.

PARTICIPATING SCIENTIFIC PERSONNEL

In addition to the Principal Investigator, Dr. Azer Bestavros, this project has supported the research of 2 current doctoral students and the research of 2 former masters students.

Doctoral Students Sue Nagy is finishing up her Ph.D. thesis on admission control and overload management for real-time databases. After defending her thesis in January 1997, she will be working with the Open Software Foundation (OSF). Paul Dell is currently working on his PhD on Data Warehousing with an emphasis on Scientific applications. He is expected to defend his thesis proposal in December 1997.

Masters Students Benjamin Mandler obtained his M.A. in June 1995. His thesis was on building a testbed for real-time databases for the study of concurrency control protocols. He is currently working with IBM in Israel. Agnes Lee completed her Masters degree in June 1996. Her thesis was on caching and prefetching algorithms for broadcast disks. She is currently working as a consultant.

References

[1] Robert Abbott and Hector Garcia-Molina. Scheduling real-time transactions: A performance evaluation. In *Proceedings of the 14th International Conference on Very Large Data Bases*, pages 1–12, Los Angeles, Ca, 1988.

[2] R. Agrawal, M. Carey, and M. Livny. Concurency control performance modeling: Alternatives and implications. *ACM Transaction on Database Systems*, 12(4), December 1987.

[3] C. Boksenbaum, M. Cart, J. Ferrié, and J. Francois. Concurrent certifications by intervals of timestamps in distributed database systems. *IEEE Transactions on Software Engineering*, pages 409–419, April 1987.

[4] A. P. Buchmann, D. C. McCarthy, M. Hsu, and U. Dayal. Time-critical database scheduling: A framework for integrating real-time scheduling and concurrency controls. In *Proceedings of the 5th International Conference on Data Engineering*, Los Angeles, California, February 1989.

[5] K. P. Eswaran, J. N. Gray, R. A. Lorie, and I. L. Traiger. The notions of consistency and predicate locks in a database system. *Communications of the ACM*, 19(11):624–633, November 1976.

[6] Peter Franaszek and John Robinson. Limitations of concurrency in transaction processing. *ACM Transactions on Database Systems*, 10(1), March 1985.

[7] J. N. Gray, R. A. Lorie, G. R. Putzolu, and I. L. Traiger. Granularity of locks and degrees of consistensy in a shared data base. In G. M. Nijssen, editor, *Modeling in Data Base Management Systems*, pages 365–395. North-Holland, Amsterdam, The Netherlands, 1976.

[8] Jayant R. Haritsa, Michael J. Carey, and Miron Livny. Dynamic real-time optimistic concurrency control. In *Proceedings of the 11th Real-Time Systems Symposium*, December 1990.

[9] Jayant R. Haritsa, Michael J. Carey, and Miron Livny. On being optimistic about real-time constraints. In *Proceedings of the 1990 ACM PODS Symposium*, April 1990.

[10] J. Huang, J. A. Stankovic, D. Towsley, and K. Ramamritham. Experimental evaluation of real-time transaction processing. In *Proceedings of the 10th Real-Time Systems Symposium*, December 1989.

[11] Jiandong Huang, John A. Stankovic, Krithi Ramamritham, and Don Towslwy. Experimental evaluation of real-time optimistic concurrency control schemes. In *Proceedings of the 17th International Conference on Very Large Data Bases*, Barcelona, Spain, September 1991.

[12] Woosaeng Kim and Jaideep Srivastava. Enhancing real-time DBMS performance with multiversion data and priority based disk scheduling. In *Proceedings of the 12th Real-Time Systems Symposium*, December 1991.

[13] Henry Korth. Triggered real-time databases with consistency constraints. In *Proceedings of the 16th International Conference on Very Large Data Bases*, Brisbane, Australia, 1990.

[14] H. Kung and John Robinson. On optimistic methods for concurrency control. *ACM Transactions on Database Systems*, 6(2), June 1981.

[15] Yi Lin and Sang Son. Concurrency control in real-time databases by dynamic adjustment of serialization order. In *Proceedings of the 11th Real-Time Systems Symposium*, December 1990.

[16] Lui Sha, R. Rajkumar, and J. Lehoczky. Concurrency control for distributed real-time databases. *ACM, SIGMOD Record*, 17(1):82–98, 1988.

[17] Lui Sha, R. Rajkumar, Sang Son, and Chun-Hyon Chang. A real-time locking protocol. *IEEE Transactions on Computers*, 40(7):793–800, 1991.

[18] Mukesh Singhal. Issues and approaches to design real-time database systems. *ACM, SIGMOD Record*, 17(1):19–33, 1988.

[19] S. Son, S. Park, and Y. Lin. An integrated real-time locking protocol. In *Proceedings of the IEEE International Conference on Data Engineering*, Tempe, AZ, February 1992.

[20] John Stankovic and Wei Zhao. On real-time transactions. *ACM, SIGMOD Record*, 17(1):4–18, 1988.